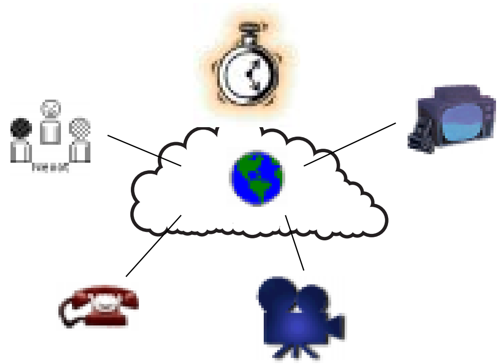
The Future of SIP

Henning Schulzrinne
Internet Real-Time Lab
Columbia University
hgs@cs.columbia.edu



SIP 2000, Paris, May 12, 2000

(Joint work with Jonathan Lennox, Gautam Nair, Ping Pan, Jonathan Rosenberg, Kundan Singh, Elin Wedlund, Xin Wang, and Jianqi Yin)

Overview

- What is SIP good at?
- SIP and Corba
- on-going IETF SIP efforts
- SIP for presence and events

Principal IETF VoIP protocols

RTP/RTCP: data transport and QoS feedback

SIP: call setup

SDP: session/media description

enum: (DNS) E.164 $\longrightarrow$ URLs

TRIP: finding “cheap” PSTN gateways, BGP-like

RTSP: voice mail, announcements

What is SIP good at?

- session setup = “out of band”
- resource location via location-independent identifier (“user@domain”, tel)
- particularly if location varies rapidly or filtering is needed (i.e., is inappropriate for DNS and LDAP)
- real-time: faster than email
- reach multiple end point simultaneously or in sequence = *forking*
- possibly hide end-point location
- delayed final answer (“ringing”) $\longleftrightarrow$ RTSP

What is SIP not meant for?

- bulk transport: media streams, files, pictures, ...
- asynchronous messaging ("email")
- resource reservation
- high-efficiency general-purpose RPC

SIP and Corba

	SIP	Corba
data	optional fields	versioning hard
	two-level hierarchy	general, C-like
hiding	dynamic	directory-based
multiple	forking proxy	no
transport	UDP, TCP, . . .	TCP
strength	inter-domain	inter-domain
generality	session set-up	RPC, events, . . .

SIP servers can benefit from Corba *locally* for user location and service creation

Current SIP efforts

- SIP to Draft Standard
- QoS and security preconditions
- inter-domain AAA and billing
- session timer for liveness detection
- early media (PSTN announcements)
- SIP for presence / instant messaging
- SIP-H.323 interworking
- reliable provisional responses
- DHCP configuration for finding SIP servers
- SIP for firewalls and NATs
- caller preferences
- services (transfer, multiparty calls, home)
- ISUP carriage

The near future of SIP

- move "RFC2543bis" to draft standard: clarifications, bug fixes
- SIP bake-off interoperability events:

	host	when	companies
1	Columbia University	April 1999	16
2	pulver.com	August 1999	15
3	Ericsson	December 1999	26
4	3Com	April 2000	36
5	pulver.com	August 2000	
6	dynamicsoft	December 2000	

SIP extensions

About 50 SIP-related current Internet drafts. . .

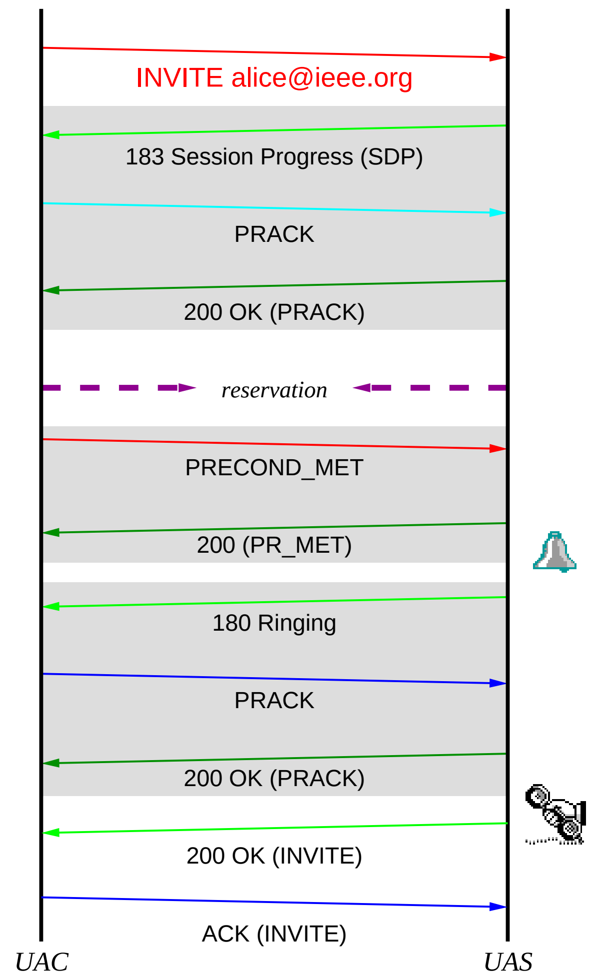
extension	example	negotiation
method	SUBSCRIBE, NOTIFY	OPTIONS, Allow
header	Session-Expires	Require
body	ISUP	Accept, Accept-*
status	183	treat by class

How to maintain backward compabitibility?

- can add ignore-if-you-like headers
- clients require capabilities via Require
- servers decline via Unsupported
- servers indicate capabilities via Supported

Coupling of signaling and QoS

- traditional (H.323) approach: use signaling to set up QoS
- but: separation of signaling and data flow
- SIP approach: security and QoS *preconditions*
- SDP in INVITE: “must have QoS”, don’t ring
- use any reservation protocol, then send PR-MET



SIP for single-line phones

- PacketCable: VoIP for cable modems
- primary line service
- NCS and DCS (distributed call signaling)
- DCS is based on SIP

- extensions:
  - “certified” caller ID
  - distributed call state
  - billing
  - media authorization

Event notification

- SUBSCRIBE to events (state changes) and then get NOTIFY
- originally, for fax events in PINT
- generic: “message waiting”, people presence, alarms . . .
- details remain to be worked out:
  - event as signaling or media stream?
  - how to describe events – headers, XML, . . . ?
  - state storage independent of user agents?
  - retrieve current state or just transitions?

Call control

Several sub-problems:

- transfer calls
  - = ask somebody to call a third party
  - originally, Also in BYE
  - proposal: TRANSFER
  - alternatives: PLEASE, with full SIP message in body or header
- multi-party call meshes
  - complicated due to distributed state
  - race conditions
  - not always needed: MCU, multicast
- third-party control $\longrightarrow$ no extensions needed

Invisible Internet telephony

- currently: stand-alone application or PSTN phone
- chat applications
- distributed games
- virtual reality environments
- web pages and applets
- links in email messages

Integrating VoIP with the web

Everything linked together, with SIP redirecting and registering:

- tel: URLs
- email: send SIP via email, redirect calls to email
- web: links to and actual content (HTML, XML, audio clips, . . .)
- chat and presence
- RTSP

Mobile Internet telephony

- user and terminal mobility are related
- mobile applications: mostly UDP (DNS, multicast) or short TCP transactions (SMTP, POP, IMAP)
- should make applications restartable
- little mobile-IP deployment
- use SIP to support mobile multimedia applications
- mobile IP and SIP mobility are complementary

Conclusion

- major protocol pieces in place
- operational issues: 911, anonymity, billing, OSS for services, ...
- other uses: controlling toasters, light switches, ...
- not just replicating existing telephone architecture and service
- programmability key to web success
- should become an invisible service
- need to keep low-end devices in mind (IPsec!)